

LAPORAN HASIL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

JOBSHEET 3



OLEH :

IVAN RIZAL AHMADI

NIM. 2341760128

SIB-1F/13

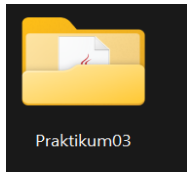
D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

3.2.1 Langkah-langkah Percobaan

1. Buat folder baru dengan nama Praktikum03.



2. Buat Class Persegi Panjang`

```
public class PersegiPanjang {  
    public int panjang;  
    public int lebar;  
}
```

3. Buat class PersegiPanjangDemo kemudian tambahkan fungsi main sebagai berikut

```
public static void main(String[] args) {  
    PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];  
}
```

4. Kemudian isikan masing masing atribut nya

```
public static void main(String[] args) {  
    PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];  
  
    arrayOfPersegiPanjang[0] = new PersegiPanjang();  
    arrayOfPersegiPanjang[0].panjang = 110;  
    arrayOfPersegiPanjang[0].lebar = 30;  
  
    arrayOfPersegiPanjang[1] = new PersegiPanjang();  
    arrayOfPersegiPanjang[1].panjang = 80;  
    arrayOfPersegiPanjang[1].lebar = 40;  
  
    arrayOfPersegiPanjang[2] = new PersegiPanjang();  
    arrayOfPersegiPanjang[2].panjang = 100;  
    arrayOfPersegiPanjang[2].lebar = 20;  
}
```

5. Cetak ke layar semua atribut dari objek **ppArray**:

```
System.out.println("Persegi panjang ke-1, panjang: " + arrayOfPersegiPanjang[0].panjang + ", lebar: "  
    + arrayOfPersegiPanjang[0].lebar);  
System.out.println("Persegi panjang ke -2, panjang: " + arrayOfPersegiPanjang[1].panjang + ", lebar: "  
    + arrayOfPersegiPanjang[1].lebar);  
System.out.println("Persegi panjang ke-3, panjang: " + arrayOfPersegiPanjang[2].panjang + ", lebar: "  
    + arrayOfPersegiPanjang[2].lebar);
```

3.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Persegi panjang ke-1, panjang: 110, lebar: 30  
Persegi panjang ke-2, panjang: 80, lebar: 40  
Persegi panjang ke-3, panjang: 100, lebar: 20
```

```
Persegi panjang ke-1, panjang: 110, lebar : 30  
Persegi panjang ke -2, panjang: 80, lebar: 40  
Persegi panjang ke-3, panjang: 100, lebar: 20  
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```

3.2.3 Pertanyaan

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method? Jelaskan! = Tidak, class yang akan dibuat sebagai array of objects tidak selalu harus memiliki atribut dan method. Sebuah class dapat memiliki hanya atribut, hanya method, atau keduanya
2. Apa yang dilakukan oleh kode program berikut?

```
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
```

Kode ini membuat sebuah array dari objek PersegiPanjang dengan nama arrayOfPersegiPanjang. Array tersebut dapat menampung tiga objek PersegiPanjang. Namun, pada saat pembuatan array ini, objek-objeknya belum diciptakan. Hanya referensi untuk objek yang dibuat, dan nilai awal atribut objek tersebut akan berupa nilai default (0 untuk tipe data int).

3. Apakah class PersegiPanjang memiliki konstruktor? Jika tidak, kenapa bisa dilakukan pemanggilan konstruktor pada baris program berikut? = Pada baris program tersebut, new PersegiPanjang() memanggil konstruktor default yang secara otomatis dibuat oleh Java, dan objek hasil instansiasi tersebut kemudian disimpan di dalam array

```
arrayOfPersegiPanjang[0] = new PersegiPanjang();
```

4. Apa yang dilakukan oleh kode program berikut?

```
arrayOfPersegiPanjang[0].panjang = 110;  
arrayOfPersegiPanjang[0].lebar = 30;
```

= Kode ini mengatur nilai atribut panjang dan lebar dari objek PersegiPanjang yang berada pada indeks ke-0 dari array arrayOfPersegiPanjang. Dengan kata lain, panjang diatur menjadi 110 dan lebar diatur menjadi 30 untuk objek tersebut.

4. Mengapa class PersegiPanjang dan PersegiPanjangDemo dipisahkan pada uji coba 3.2?
= Karena class PersegiPanjangDemo berfungsi sebagai entry point untuk menjalankan dan menampilkan informasi atau uji coba terkait class PersegiPanjang. Pemisahan ini meningkatkan keterbacaan, pemahaman, dan pemeliharaan kode.

3.3.1 Langkah-langkah Percobaan

1. Import scanner pada class PersegiPanjangDemo.

```
import java.util.Scanner;
```

Note: Letakkan kode import dibawah kode package (jika ada).

2. Pada praktikum 3.2 poin nomor 4, modifikasi kode program sebagai berikut. Buat objek **Scanner** untuk menerima input, kemudian lakukan looping untuk menerima informasi panjang dan lebar:

```
Scanner sc = new Scanner(System.in);
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];

for (int i = 0; i < 3; i++) {
    arrayOfPersegiPanjang[i] = new PersegiPanjang();

    System.out.println("Persegi panjang ke-" + (i + 1));
    System.out.print(s:"Masukkan panjang: ");
    arrayOfPersegiPanjang[i].panjang = sc.nextInt();
    System.out.print(s:"Masukkan lebar: ");
    arrayOfPersegiPanjang[i].lebar = sc.nextInt();
}
```

3. Pada praktikum 3.2 poin nomor 5, modifikasi kode program sebagai berikut. Lakukan looping untuk mengakses elemen array dan menampilkan informasinya ke layar:

```
// Use a different variable name for the inner loop
for (int i = 0; i < 3; i++) {
    System.out.println("Persegi panjang ke-" + (i + 1) + ", panjang: "
        + arrayOfPersegiPanjang[i].panjang + ", lebar: " + arrayOfPersegiPanjang[i].lebar);
}
```

Hasil running :

```
Persegi panjang ke-1
Masukkan panjang: 5
Masukkan lebar: 4
Persegi panjang ke-2
Masukkan panjang: 3
Masukkan lebar: 4
Persegi panjang ke-3
Masukkan panjang: 5
Masukkan lebar: 6
Persegi panjang ke-1, panjang: 5, lebar: 4
Persegi panjang ke-2, panjang: 3, lebar: 4
Persegi panjang ke-3, panjang: 5, lebar: 6
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```

3.3.3 Pertanyaan

1. Tambahkan method cetakInfo() pada class PersegiPanjang kemudian modifikasi kode program pada langkah no 3.

```
arrayOfPersegiPanjang[i].cetakInfo();
```

2. Misalkan Anda punya **array baru** bertipe array of PersegiPanjang dengan nama **myArrayOfPersegiPanjang**. Mengapa kode berikut menyebabkan error? = karena saat pembuatan array myArrayOfPersegiPanjang, hanya referensi objek untuk setiap elemen array yang dibuat, tetapi objek-objek itu sendiri belum diciptakan.

```
PersegiPanjang[] myArrayOfPersegiPanjang = new PersegiPanjang[100];  
myArrayOfPersegiPanjang[5].panjang = 20;
```

3.4 Constructor Berparameter

Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

1. Pada class PersegiPanjang, tambahkan constructor berparameter

```
// Constructor berparameter  
public PersegiPanjang(int p, int l) {  
    panjang = p;  
    lebar = l;  
}
```

2. Run program kemudian amati hasilnya.

```
Persegi panjang ke-1, panjang: 110, lebar : 30  
Persegi panjang ke -2, panjang: 80, lebar: 40  
Persegi panjang ke-3, panjang: 100, lebar: 20  
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```

3. Modifikasi class PersegiPanjangDemo sehingga instansiasi dilakukan menggunakan constructor berparameter

```

Scanner sc = new Scanner(System.in);
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
int panjang, lebar;

for (int i = 0; i < 3; i++) {
    System.out.println("Persegi panjang ke-" + (i + 1));
    System.out.print(s:"Masukkan panjang: ");
    panjang = sc.nextInt();
    System.out.print(s:"Masukkan lebar: ");
    lebar = sc.nextInt();
    PersegiPanjang[] arrayOfPersegiPanjang - PersegiPanjangDemo.main(String[])
    arrayOfPersegiPanjang[i] = new PersegiPanjang(panjang, lebar);
}

```

3.4.2 Verifikasi Percobaan

Hasil running :

Hasil

```

Persegi panjang ke-1
Masukkan panjang: 3
Masukkan lebar: 4
Persegi panjang ke-2
Masukkan panjang: 3
Masukkan lebar: 4
Persegi panjang ke-3
4
Persegi panjang ke-2, panjang: 3, lebar: 4
Persegi panjang ke-3, panjang: 3, lebar: 4
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>

```

3.4.3 Pertanyaan

1. Mengapa terjadi compile error pada langkah no 2? = Compile error pada langkah no 2 terjadi karena terdapat karakter yang tidak valid
2. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya = Ya, suatu class dapat memiliki lebih dari satu constructor. Ini dikenal sebagai overloading constructor

```

public class PersegiPanjang {
    public int panjang;
    public PersegiPanjang.PersegiPanjang()
    public PersegiPanjang() {
    }

    // Constructor berparameter
    public PersegiPanjang(int p, int l) {
        panjang = p;
        lebar = l;
    }

    public void cetakInfo() {
        System.out.println("Persegi panjang, panjang: " + panjang + ", lebar: " + lebar);
    }

    Run|Debug
    public static void main(String[] args) {

```

3. Tambahkan method `hitungLuas()` dan `hitungKeliling()` =

```
public int hitungLuas() {  
    return panjang * lebar;  
}  
  
// Method untuk menghitung keliling persegi panjang  
public int hitungKeliling() {  
    return 2 * (panjang + lebar);  
}
```

4. Hasil modifikasi =

```
Persegi panjang ke-1  
Masukkan panjang: 2  
Masukkan lebar: 3  
Persegi panjang ke-2, panjang: 2, lebar: 3, luas: 6, keliling: 10  
Persegi panjang ke-2  
Masukkan panjang: 2  
Masukkan lebar: 3  
Persegi panjang ke-2, panjang: 2, lebar: 3, luas: 6, keliling: 10  
Persegi panjang ke-3  
Masukkan panjang: 2  
Masukkan lebar: 3  
Persegi panjang ke-2, panjang: 2, lebar: 3, luas: 6, keliling: 10  
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```

5. Modifikasi kode program pada praktikum 3.3 agar length array ditentukan dari user melalui input dengan Scanner =

```
for (int i = 0; i < jumlahPersegiPanjang; i++) {  
    System.out.println("Persegi panjang ke-" + (i + 1));  
    System.out.print(s:"Masukkan panjang: ");  
    int panjang = sc.nextInt();  
    System.out.print(s:"Masukkan lebar: ");  
    int lebar = sc.nextInt();  
  
    arrayOfPersegiPanjang[i] = new PersegiPanjang(panjang, lebar);  
    arrayOfPersegiPanjang[i].cetakInfo();  
}
```

Dalam modifikasi ini, program meminta pengguna untuk memasukkan jumlah persegi panjang (`jumlahPersegiPanjang`). Kemudian, array `arrayOfPersegiPanjang` dibuat dengan panjang sesuai dengan input pengguna. Selanjutnya, proses input dan cetak info diulangi sebanyak jumlah yang dimasukkan oleh pengguna.

Lalu untuk hasil running program adalah seperti berikut

=

```
Persegi panjang ke-1, panjang: 110, lebar: 30  
Persegi panjang ke -2, panjang: 80, lebar: 40  
Persegi panjang ke-3, panjang: 100, lebar: 20  
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03> c:: cd 'c:\Users\ivanr\OneDrive\Desktop\Praktikum03'; & 'C:\Program Files\Java\  
: 30  
Persegi panjang ke -2, panjang: 80, lebar: 40  
Persegi panjang ke-3, panjang: 100, lebar: 20  
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```

3.5 Tugas

1. Sebuah kampus membutuhkan program untuk menampilkan informasi mahasiswa berupa nama, NIM, jenis kelamin dan juga IPK mahasiswa. Program dapat menerima input semua informasi tersebut, kemudian menampilkanya kembali ke user. Tambahkan informasi rata-rata IPK pada bagian akhir. Gunakan looping dengan **FOR** untuk pembuatan object. Gunakan looping dengan **FOREACH** untuk pencetakan data ke layar.

= Kelas Mahasiswa:

Atribut: Kelas ini memiliki empat atribut private: nama, nim, jenisKelamin, dan ipk.

Constructor: Terdapat satu constructor yang digunakan untuk menginisialisasi objek Mahasiswa dengan nilai yang diberikan saat pembuatan objek.

Method cetakInfo(): Method ini mencetak informasi lengkap mengenai mahasiswa ke layar.

Method getIpk(): Method ini mengembalikan nilai IPK dari objek Mahasiswa.

Kelas MahasiswaDemo (Main Class):

Method main(): Pada method ini, program mengambil input dari pengguna untuk tiga mahasiswa, membuat objek Mahasiswa untuk setiap input, menyimpan objek-

objek tersebut dalam array, mencetak informasi mahasiswa, dan menghitung rata-rata IPK.

Array arrayMahasiswa: Digunakan untuk menyimpan objek-objek Mahasiswa.

```
1  import java.util.Scanner;
2
3  public class MahasiswaDemo {
4
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7
8          int jumlahMahasiswa = 3; // Jumlah mahasiswa yang akan dimasukkan
9
10         Mahasiswa[] arrayMahasiswa = new Mahasiswa[jumlahMahasiswa];
11
12         // Input data mahasiswa
13         for (int i = 0; i < jumlahMahasiswa; i++) {
14             System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1));
15
16             System.out.print("Masukkan nama: ");
17             String nama = sc.next();
18
19             System.out.print("Masukkan NIM: ");
20             String nim = sc.next();
21
22             System.out.print("Masukkan jenis kelamin (L/P): ");
23             char jenisKelamin = sc.next().charAt(0);
24
25             System.out.print("Masukkan IPK: ");
26             double ipk = sc.nextDouble();
27
28             arrayMahasiswa[i] = new Mahasiswa(nama, nim, jenisKelamin, ipk);
29         }
30
31         // Cetak data mahasiswa
32         for (int i = 0; i < jumlahMahasiswa; i++) {
33             System.out.println("\nData Mahasiswa ke-" + (i + 1));
34             arrayMahasiswa[i].cetakInfo();
35         }
36
37         // Hitung rata-rata IPK
38         double totalIpk = 0;
39         for (Mahasiswa mahasiswa : arrayMahasiswa) {
40             totalIpk += mahasiswa.getIpk();
41         }
42         double rataRataIpk = totalIpk / jumlahMahasiswa;
43
44         System.out.println("\nRata-rata IPK: " + rataRataIpk);
45     }
46 }
47
48 class Mahasiswa {
49     private String nama;
50     private String nim;
51     private char jenisKelamin;
52     private double ipk;
53
54     public Mahasiswa(String nama, String nim, char jenisKelamin, double ipk) {
55         this.nama = nama;
56         this.nim = nim;
57         this.jenisKelamin = jenisKelamin;
58         this.ipk = ipk;
59     }
60
61     public void cetakInfo() {
62         System.out.println("Nama: " + nama);
63         System.out.println("Nim: " + nim);
64         System.out.println("Jenis kelamin: " + jenisKelamin);
65         System.out.println("Nilai IPK: " + ipk);
66     }
67
68     public double getIpk() {
69         return ipk;
70     }
71 }
72 }
```


Looping untuk Input Mahasiswa: Menggunakan loop for untuk menerima input data mahasiswa sebanyak tiga kali.

Looping untuk Cetak Data Mahasiswa: Menggunakan loop for dan enhanced for (foreach) untuk mencetak informasi mahasiswa yang telah dimasukkan.

Perhitungan Rata-rata IPK: Menggunakan loop foreach untuk menjumlahkan semua IPK mahasiswa dan menghitung rata-rata IPK.

Input dan Output: Program ini meminta pengguna untuk memasukkan nama, NIM, jenis kelamin, dan IPK untuk setiap mahasiswa. Kemudian, program mencetak informasi mahasiswa dan rata-rata IPK ke layar.

Scanner: Digunakan untuk menerima input dari pengguna.

Hasil Running Program =

```
Masukkan data mahasiswa ke-1
Masukkan nama: ijiji
Masukkan NIM: 34232
Masukkan jenis kelamin (L/P): 1
Masukkan IPK: 3

Masukkan data mahasiswa ke-2
Masukkan nama: deded
Masukkan NIM: 232323
Masukkan jenis kelamin (L/P): 1
Masukkan IPK: 4

Masukkan data mahasiswa ke-3
Masukkan nama: mjuejd
Masukkan NIM: 232323
Masukkan jenis kelamin (L/P): 1
Masukkan IPK: 4

Data Mahasiswa ke-1
Nama: ijiji
Nim: 34232
Jenis kelamin: 1
Nilai IPK: 3.0

Data Mahasiswa ke-2
Nama: deded
Nim: 232323
Jenis kelamin: 1
Nilai IPK: 4.0

Data Mahasiswa ke-3
Nama: mjuejd
Nim: 232323
Jenis kelamin: 1
Nilai IPK: 4.0

Rata-rata IPK: 3.6666666666666665
```

2. Implementasikan tugas ASD Teori no 1 ke dalam kode program. Buatlah array of objects berdasarkan input dari user dan tampilkan data seperti pada Tugas no 1. =

```
Run | Debug
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    for (int i = 0; i < 3; i++) {
        System.out.println("\nMasukkan data pemesanan ke-" + (i + 1));

        System.out.print(s:"Apakah anda Mahasiswa Polinema? (Iya/Tidak) ");
        input.next();
        PemesananRuang pemesanan = new PemesananRuang();

        if (answer.equalsIgnoreCase(anotherString:"iya")) {
            System.out.print(s:"Masukkan Nama Pemesan: ");
            pemesanan.setNamaMahasiswa(input.next());
            System.out.print(s:"Masukkan NIM Pemesan: ");
            pemesanan.setNimMahasiswa(input.next());
        } else {
```

Penggunaan PemesananRuangan Class:

Program memiliki kelas PemesananRuangan yang memiliki variabel-variabel instance untuk menyimpan informasi pemesanan seperti nama mahasiswa, NIM mahasiswa, nama ruangan, jumlah orang, jumlah jam, dan tanggal pemesanan.

Looping for untuk Input Data:

Dengan menggunakan loop for, program memungkinkan pengguna untuk menginputkan data pemesanan untuk beberapa kali pesanan (dalam contoh ini, tiga kali).

Pemilihan Mahasiswa atau Non-Mahasiswa:

Program menanyakan apakah pengguna adalah mahasiswa Polinema atau bukan. Jika iya, maka program meminta informasi tambahan seperti NIM.

Setter Method:

Program menggunakan metode setter seperti setNameMahasiswa, setNimMahasiswa, setNameRuangan, dll., untuk mengatur nilai variabel instance dari kelas.

Metode cetakInfo:

Ada metode cetakInfo yang digunakan untuk mencetak detail pemesanan ke layar.

Hasil Running Program

```
=====
                        Detail Pemesanan
=====
Nama Mahasiswa: rffrf
NIM Mahasiswa: 34344
Nama Ruangan: ffvv
Jumlah Orang: 34 orang
Jumlah Jam yang Dipesan: 4.0 jam
Tanggal Pemesanan: 34/33/2222

Masukkan data pemesanan ke-3
Apakah anda Mahasiswa Polinema? (Iya/Tidak) iya
Masukkan Nama Pemesan: 3rfr
Masukkan NIM Pemesan: 3443434
Masukkan Nama Ruangan: 3434
Masukkan Jumlah Orang: 34343
Masukkan Jumlah Jam yang ingin dipesan: 34
Masukkan Tanggal Pemesanan (dd/mm/yyyy): 343434

=====
                        Detail Pemesanan
=====
Nama Mahasiswa: 3rfr
NIM Mahasiswa: 3443434
Nama Ruangan: 3434
Jumlah Orang: 34343 orang
Jumlah Jam yang Dipesan: 34.0 jam
Tanggal Pemesanan: 343434
PS C:\Users\ivanr\OneDrive\Desktop\Praktikum03>
```